



**UNIX<sup>TM</sup> System**  
**Release Description**  
**System V**

January 1983

UNIX is a trademark of  
Bell Telephone Laboratories, Incorporated.

Copyright©1983 Western Electric Company, Incorporated.

Portions of this document were copyrighted  
1979 Bell Telephone Laboratories, Incorporated.  
1980 Western Electric Company, Incorporated.

This document was prepared with specific references to use of the UNIX<sup>1</sup> System on a particular processor, the Western Electric Company 3B20S, which is not presently available except for internal use within the Bell System. However, the information contained herein is generally applicable to use of the UNIX System on various processors which are available in the general trade.

The *vi (ex)* editor is based on software developed by the University of California, Berkeley, California, Computer Science Division, Department of Electrical Engineering and Computer Science, and such software is owned and licensed by the Regents of the University of California.

<sup>1</sup> UNIX is a trademark of  
Bell Telephone Laboratories, Incorporated.

DEC, PDP, VAX, UNIBUS, MASSBUS, DECnet, and SBI  
are trademarks of Digital Equipment Corporation.

HYPERchannel is a trademark of Network Systems Corporation.

Ethernet is a trademark of Xerox Corporation.

## CONTENTS

1. INTRODUCTION . . . . .	1
1.1 Achievements . . . . .	1
1.2 Scope . . . . .	1
2. RELEASE OVERVIEW . . . . .	1
2.1 Release Features . . . . .	2
2.2 Documentation . . . . .	3
2.3 Transition Aids . . . . .	3
3. PERFORMANCE . . . . .	3
3.1 C Language . . . . .	3
3.2 Kernel Operations . . . . .	5
3.3 Terminal Handling . . . . .	5
3.4 Simulated Timesharing Workload . . . . .	8
3.5 Administrative Guidelines . . . . .	9
4. DOCUMENTATION . . . . .	10
4.1 Published Documents . . . . .	10
4.2 Interim Documents . . . . .	11
5. ITEMS NO LONGER SUPPORTED . . . . .	12
5.1 DQS and Fget . . . . .	12
5.2 Floating Point Simulation . . . . .	12
5.3 DEC Processors . . . . .	12
6. TRANSITIONAL INFORMATION . . . . .	12
6.1 Release Tapes . . . . .	12
6.2 Installation Documentation . . . . .	12
6.3 File System Conversion (VAX only) . . . . .	13
6.4 Init & Getty Interface . . . . .	13
6.5 C Programs . . . . .	13
6.6 Vi . . . . .	13
6.7 Uucp . . . . .	13
6.8 PDP Size Limitations . . . . .	14
6.9 Modification Requests . . . . .	14
7. KNOWN DEFICIENCIES . . . . .	14
7.1 Compilers and Language Tools . . . . .	14
7.2 Communications . . . . .	15
7.3 Lp . . . . .	16
7.4 Init and Getty . . . . .	16
7.5 File . . . . .	16
7.6 Du and Ls . . . . .	16
7.7 /etc/checkall . . . . .	17
7.8 Mm Macros . . . . .	17
7.9 Games . . . . .	17
8. DISCLAIMERS . . . . .	18
APPENDICES: . . . . .	19
Appendix A: New Pages for the <i>UNIX System User's Manual - System V</i> and the <i>UNIX System Administrator's Manual - System V</i>	
Appendix B: Setting Up the UNIX System (DEC)	
Appendix C: System V Init and Getty	
Appendix D: An Introduction to Display Editing with Vi	

**System V**

**Appendix E: Ex Reference Manual**

**Appendix F: Uucp Tutorial**

**Appendix G: Uucp Administrators Manual**

**Appendix H: Installing an Auto Call Facility on a UNIX System**

**Appendix I: UNIX System Modification Requests Resolved in UNIX System V**

LIST OF FIGURES

Figure 1. Terminal Output - Raw Mode . . . . .	6
Figure 2. Terminal Output - Cooked Mode . . . . .	7
Figure 3. Terminal Data Loop-Around - Raw Mode . . . . .	8
Figure 4. Terminal Data Loop-Around - Cooked Mode . . . . .	9
Figure 5. UNIX System V Throughput vs. Applied Load . . . . .	10

LIST OF TABLES

TABLE 1. Normalized Object Code Execution Speed . . . . .	4
TABLE 2. Normalized Object Code Text Size . . . . .	4
TABLE 3. Normalized Compilation Speed . . . . .	4
TABLE 4. Kernel Function Execution Time in msec (III time / V time) . . . . .	5
TABLE 5. Peak Output Rate in kilobaud . . . . .	6
TABLE 6. Maximum Loop-Around Rate in kilobaud . . . . .	7
TABLE 7. Peak Throughput in 1000 jobs/hour . . . . .	8
TABLE 8. State of /usr/games . . . . .	17

## 1. INTRODUCTION

*UNIX*<sup>1</sup> *System V* is being provided for Digital Equipment Corporation (DEC)<sup>2</sup> VAX<sup>2</sup> and PDP<sup>2</sup> processors.

*UNIX System V* is provided for the following processors:

- DEC VAX-11/780 and VAX-11/750
- DEC PDP-11/70

and includes features in the following general categories:

- Kernel Enhancements and Changes
- Enhancements for DEC Processors
- Languages and Libraries
- Communication and Networking
- Commands and Administration Aids

### 1.1 Achievements

The single most noticeable achievement in this release is the significant gain in overall system performance. This improvement translates directly into improved system response time for the user, or an increased capacity in users per system.

*System V* provides UNIX System users with timely and useful documentation. The standard documentation is included as part of the release and is automatically distributed with the software.

### 1.2 Scope

This *System Release Description (SRD)* contains a current list of the *System V* features, and current information on documentation and transitional (*System III* to *System V*) concerns. It is intended to provide an overall description of *System V*, while emphasizing the system performance improvements, new documentation, and transitional information.

*Section 2* provides a general overview of what is in this release and what is provided with it. A detailed performance evaluation of *System V* is given in *Section 3*. *Section 4* describes the documentation available with *System V*.

*Sections 5, 6, and 7* document items no longer supported, transitional aids, and known deficiencies, respectively. This information is intended to ease the installation of this release and the transition of software and users to *System V*. Please read these sections before attempting to install this release. Also, please read the documents referenced in these sections, as they will contain many helpful hints that will make the installation job easier.

## 2. RELEASE OVERVIEW

This section provides an overview of *System V*.

---

<sup>1</sup> UNIX is a trademark of Bell Telephone Laboratories, Incorporated.

<sup>2</sup> DEC, VAX, and PDP are trademarks of Digital Equipment Corporation.

## 2.1 Release Features

The following is a summary list of the individual features in *System V*.

### 2.1.1 Kernel Enhancements and Changes

**2.1.1.1 Operating System Performance.** The performance of the operating system on the VAX systems has been improved by a larger block file system, eliminating redundant work, and providing a larger number of I/O buffers. The new features are:

- 1 Kilobyte Block File System
- Physical I/O Buffer Facility
- Larger Buffer Pool
- File Table Free List
- Faster *fork* System Call
- Faster *open* System Call (inode hashing)
- Hashing of the *sleep/wakeup* Mechanism (also on PDP)

**2.1.1.2 File System Changes.** The file system has been enhanced to maintain the consistency of the disk information. This eliminates a class of disk errors associated with system crashes. The new features are:

- Ordered Writes (VAX only)
- Dual *fsck* Capability
- Faster *fsck* Option (VAX only)
- Improved File System Access
- *fuser* Command

**2.1.1.3 Operating System Enhancements.** *System V* provides new features including:

- Processor Type Identification
- Interprocess Communication
- *init* and *getty*
- New General Disk Driver

### 2.1.2 New Hardware Enhancements

The UNIX System running on the Digital Equipment Corporation processors has been updated to support new devices introduced by the hardware vendor. The new devices are:

- RP07 - Large Capacity (516 Megabyte) Disk Drive (VAX Only)
- TU78 - High Speed (6250 BPI) Tape Drive (VAX-11/780 Only)
- RM80 - Medium Capacity (124 Megabyte) Disk Drive (VAX Only)
- TS11 - Medium Speed (1600 BPI) Tape Drive (VAX-11/750 Only)
- ML11 - MOS Memory

### 2.1.3 Languages and Libraries

The language tools of *System V* have been changed to provide run-time efficiency, and improved performance. In particular, the C library has been changed to take advantage of the improvements in the operating system and to perform its functions more efficiently. The list of features includes:

- System Generation Software (SGS) Performance Improvements (VAX only)
- Improved Machine Language Processors (MLP) (VAX only)
- Library Performance Enhancements
- Larger Buffer Size
- Profiled Versions of the Libraries
- Math Library Enhancements
- New Library Functions

### 2.1.4 Communication and Networking

The Communications and Networking Facilities of *System V* provide enhanced support of networking across all supported processors. Kernel and networking program enhancements are



provided for:

- *uucp*
- *Common Synchronous Interface (CSI)*
- *Remote Job Entry (RJE)*

### 2.1.5 *Commands and Administration Aids*

*System V* provides the user and administrator with uniform capabilities across the supported processors. This list includes:

- Incremental Tape Back-up
- *vi/lex* Screen Editor Support
- Accounting Enhancements
- System Activity Report
- Line Printer Spooling System

### 2.2 Documentation

The new documentation highlights are:

- System Error Message Manual
- F77 library documentation

The release documentation is described in *Section 4* of this *SRD*.

### 2.3 Transition Aids

Customers are required to perform some conversion functions when installing *System V*. A guide to aid in the transition from *System III* to *System V*, *UNIX System Transition Aid - System V*, contains documents concerned with:

- Converting to a 1 Kilobyte Block File System
- UNIX System Archive Format Changes
- User Level Changes to the C Language
- The Common Object File Format
- UNIX Operating System Environment Changes

See *Sections 5, 6, and 7* for more details.

Conversion tools will be provided. These include:

- *arcv(1)* - convert archive files from PDP-11 to common archive format
- *convert(1)* - convert object and archive files to common formats
- *fsba(1)* - file system block analyzer
- *non-btl(1)* - re-install mm macros without Bell Labs specific features

## 3. PERFORMANCE

*System V* offers significantly improved performance on the 32-bit VAX family machines relative to *System III*. Although the magnitude of the improvement is dependent on workload, VAX time-sharing installations should experience an approximate increase in capacity of 26%. These gains are the result of changes in the UNIX System kernel, the C compiler and libraries.

### 3.1 C Language

The following results are based on a collection of small (about one hundred lines average) C language programs which do not use library functions. The faster compilation times reflect changes to the kernel and improvements in many C library functions.

---

<sup>1</sup> The *vi(lex)* editor is based on software developed by the University of California, Berkeley, California, Computer Science Division, Department of Electrical Engineering and Computer Science, and such software is owned and licensed by the Regents of the University of California.

### 3.1.1 Object Code Execution Speed

VAX-11/780 and PDP-11/70 object code speeds are essentially unchanged from System III. In Table 1, object code execution speed factors are normalized to the VAX-11/780 by dividing the 11/780 times by the respective times for each table entry. Larger numbers indicate faster program execution.

<i>Machine</i>	<i>Speed</i>
VAX-11/780	1.00
VAX-11/750	0.64
PDP-11/70	0.85

TABLE 1. Normalized Object Code Execution Speed

### 3.1.2 Object Code Size

The following relative size results were obtained by subtracting the size of a null program object from the sizes of the benchmark object modules and dividing by the VAX-11/780 result. In the table, smaller numbers indicate more compact storage on the respective machine. System V object code text size on the VAX-11/780 is about the same as that of System III. PDP-11/70 System V text size is slightly smaller than for System III.

<i>Machine</i>	<i>Size</i>
VAX-11/780	1.00
VAX-11/750	1.00
PDP-11/70	1.04

TABLE 2. Normalized Object Code Text Size

### 3.1.3 Compilation Speed

The following table shows compilation speed factors normalized to the VAX-11/780 System V result. Larger numbers indicate faster program execution. Compilations of small programs on the VAX machines take place about twenty percent faster using System V. Table 3 figures are rough guides only, since program size and use of library functions affect compilation times differently on the different machines.

<i>Machine</i>	<i>Speed</i>
VAX-11/780	1.00
VAX-11/750	0.65
PDP-11/70	0.73

TABLE 3. Normalized Compilation Speed

### 3.1.4 Libraries

Significant performance improvements were made in the following C library I/O functions: *getc(3S)*, *putc(3S)*, *gets(3S)*, *puts(3S)*, *fgets(3S)*, *fputs(3S)*, *fread(3S)* and *fwrite(3S)*. For small numbers of bytes handled, the improvements tend to be minor; for larger operand strings, however, the revised functions are up to a factor of five faster.

### 3.2 Kernel Operations

The operating system kernel for the VAX-11/780 is much faster and is responsible for much of the improvement observed for most commands. Table 4\* lists the times in milliseconds for a variety of kernel functions along with the improvements (in parenthesis) calculated as System III time divided by System V time.

Operation	VAX-11/780	VAX-11/750	PDP-11/70
syscall ( <i>getpid</i> )	.18 (1.33)	.30	.23 (0.96)
context switch	.71 (1.37)	1.02	1.03 (1.16)
<i>chdir</i> ''	1.22 (2.49)	1.92	3.35 (0.95)
<i>open/close</i> 'file'	2.47 (2.77)	3.85	6.82 (0.95)
search path 3 lev.	5.93 (2.94)	8.96	17.1 (0.95)
search dir. 32 pos.	3.03 (2.48)	5.11	11.1 (0.97)
pipe 512 bytes	3.21 (1.14)	5.05	5.69 (0.99)
pipe 1024 bytes	3.91 (1.30)	5.66	8.61 (1.00)
read 1 - buf cache	.94 (1.11)	1.56	1.42 (0.97)
read 512 - buf cache	1.38 (1.10)	2.05	2.51 (0.97)
read 1024 - buf cache	1.81 (1.32)	2.57	4.17 (0.99)
access disk block	3.10 (1.00)	4.69	4.19 (0.88)
read 4K file	19.4 (4.1)	34.7	66.3 (0.88)
<i>fork/exit</i> 8K data	21.6 (1.07)	28.3	23.9 (0.98)
<i>exec</i> 8K bss	18.5 (1.23)	25.7	34.6 (1.08)
<i>exec</i> 8K data	25.5 (1.41)	37.2	51.7 (1.03)
<i>exec</i> 512 arglist	61.7 (1.02)	94.8	100 (1.02)

TABLE 4. Kernel Function Execution Time in msec (III time / V time)

In Table 4, context switches are faster largely as a result of an improved *sleep/wakeup* mechanism. The faster VAX file operations are attributable to improved file and inode table searches.

The improvements in read and pipe operations for 1024 byte units on the VAX are mainly the result of the switch to the larger 1024 byte block size. Note that a single 1024 byte transfer takes place much faster than a pair of 512 byte operations; user application programs that currently do I/O in units of 512 bytes will obtain much better performance if changed to perform 1024 byte transfers. For programs that do I/O via the C library, this is taken care of automatically when the programs are recompiled. Although the time for disk block access has changed relatively little, disk accesses now transfer twice as much data; the improved performance is demonstrated by the shorter CPU times to read a 4096 byte file. The *exec* performance improvements are also due primarily to the larger block size.

### 3.3 Terminal Handling

Figures 1 through 4 and Tables 5 and 6 show the output and loop-around bandwidths for System V expressed in kilobytes per CPU second. CPU utilization for terminal handling on a production system can be derived as the ratio of observed bytes per second on the system in question to the corresponding bandwidths from Figures 1 through 4.

\* The results for *chdir(2)*, *open(2)/close(2)*, path search and directory search are for 120 inode table slots in use on the VAX-11/780 and 80 slots in use on the VAX-11/750 and PDP-11/70.

### 3.3.1 Terminal Output

The results in Figures 1 and 2 were obtained by writing data at 9600 baud from the system to twenty terminal lines. The VAX-11/780 with KMC/DZ assist yields very high bandwidths. For the VAX-11/780 without KMC/DZ and the PDP-11/70, the required CPU involvement in character processing yields lower bandwidths overall, especially in "cooked" mode. The poor showing of the VAX-11/780 without KMC/DZ compared to the PDP-11/70 results from the interrupt-per-character nature of the DZ-11 as opposed to the DMA property of the DH-11 used with the PDP-11/70.

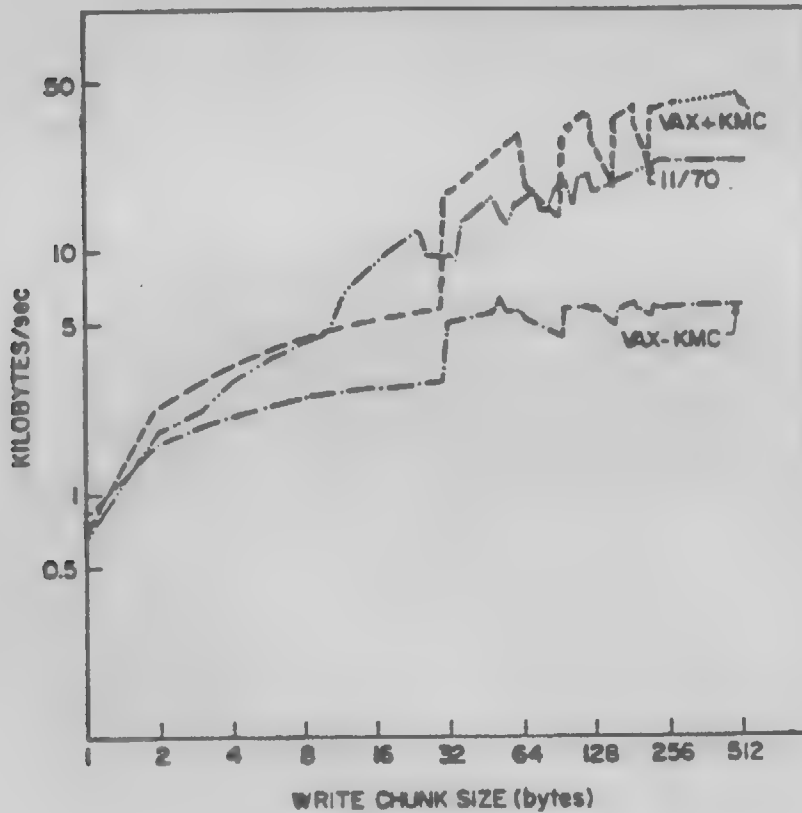


Figure 1. Terminal Output - Raw Mode

The peak output rates for System V are summarized in Table 5

Machine	Cooked	Raw
VAX-11/780+KMC	40.1	40.3
VAX-11/780-KMC	2.9	5.6
PDP-11/70	6.1	23.7

TABLE 5. Peak Output Rate in kilobaud

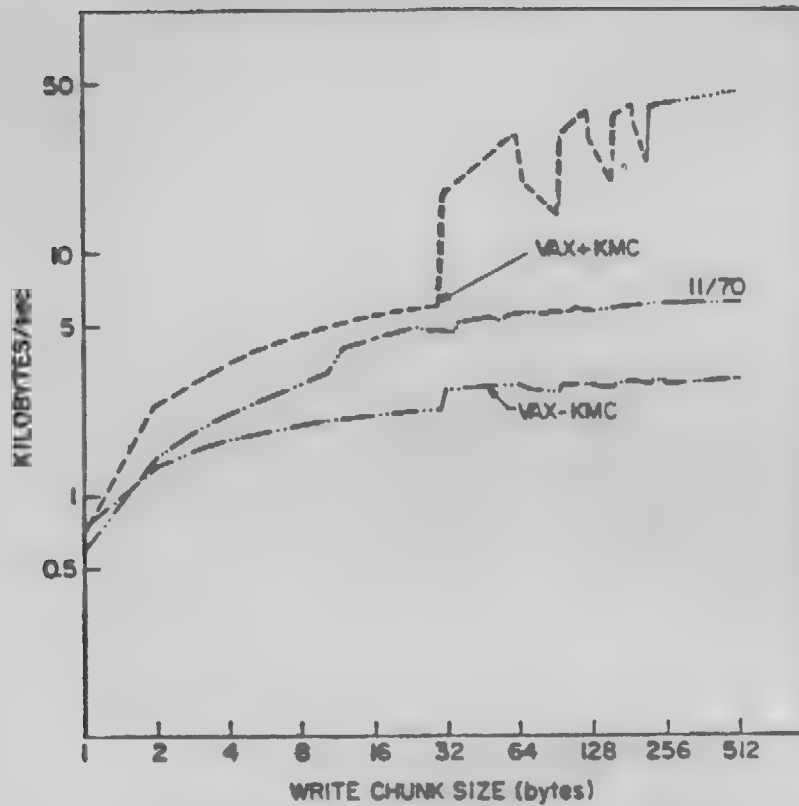


Figure 2. Terminal Output - Cooked Mode

3.3.2 Terminal Data Loop-Around

The loop-around bandwidths in Figures 3 and 4 were obtained by attaching special hardware to ten lines which circulated the outgoing data back to the input. The bandwidths in these figures represent input limitations, since the output-only bandwidths of Figures 1 and 2 are much greater.

Table 6 summarizes the maximum bandwidths obtainable for data loop-around. Bandwidths for the VAX-11/780 with KMC/DZ system are up to an order of magnitude better than for System III as the result of revision of the KMC driver, which now avoids unnecessary character-at-a-time interrupts by the KMC.

Machine	Cooked	Raw
VAX-11/780+KMC	2.4	4.2
VAX-11/780-KMC	1.1	1.7
PDP-11/70	1.5	2.3

TABLE 6. Maximum Loop-Around Rate in kilobaud

Results indicate that the VAX-11/780 with KMC and PDP-11/70 systems are unlikely to be burdened by CPU overhead from terminal handling. Therefore, no restrictions need be placed on the use of 9600 baud lines for normal interactive use.

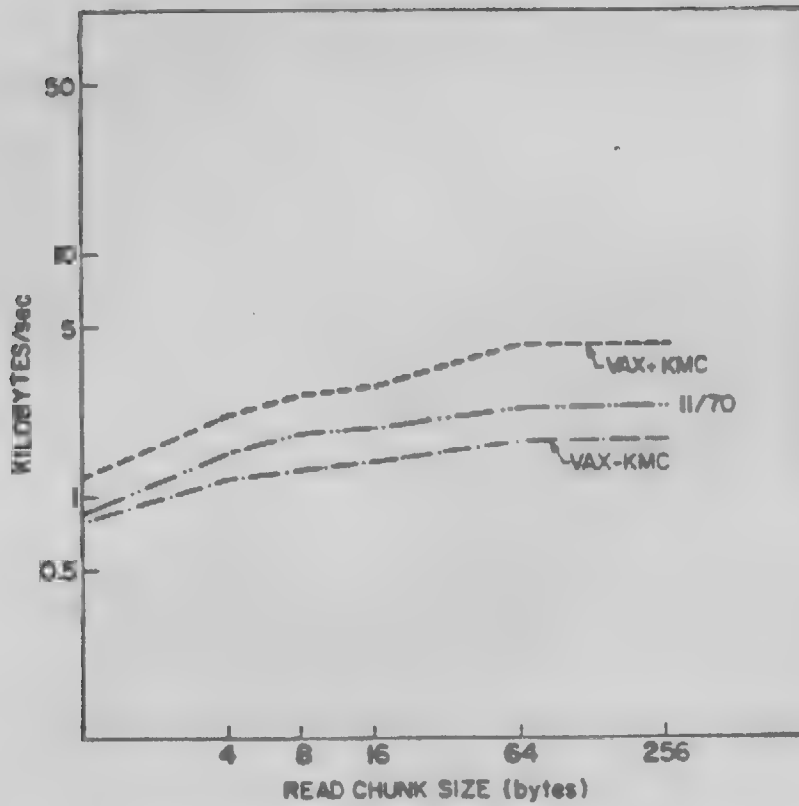


Figure 3. Terminal Data Loop-Around - Raw Mode

#### 3.4 Simulated Timesharing Workload

The time-sharing simulation benchmark used measures the throughput of the system by executing shell scripts comprised of UNIX System commands. These scripts are modeled on measurements of a program development environment. An increasing number of scripts are executed concurrently to test the system under a variety of load conditions. Each script executes about forty of the commands observed to consume the most CPU time in the time sharing environment.

The results given in Figure 5 and Table 7 are expressed as throughput in jobs per hour versus load. With light loads, the system is partially idle waiting for disk I/O. With the heavier loads, system overhead increases due to reduced buffer cache hit ratio and swapping (although the VAX machine had sufficient memory so that swapping did not occur). Estimated peak throughputs of the various systems are compared in Table 7.

Machine	III	V	(% improvement)
VAX-11/780	7.74	9.75	(+26%)
VAX-11/750	n/a	6.01	(n/a)
PDP-11/70	5.64	5.83	(-3%)

TABLE 7. Peak Throughput in 1000 jobs/hour

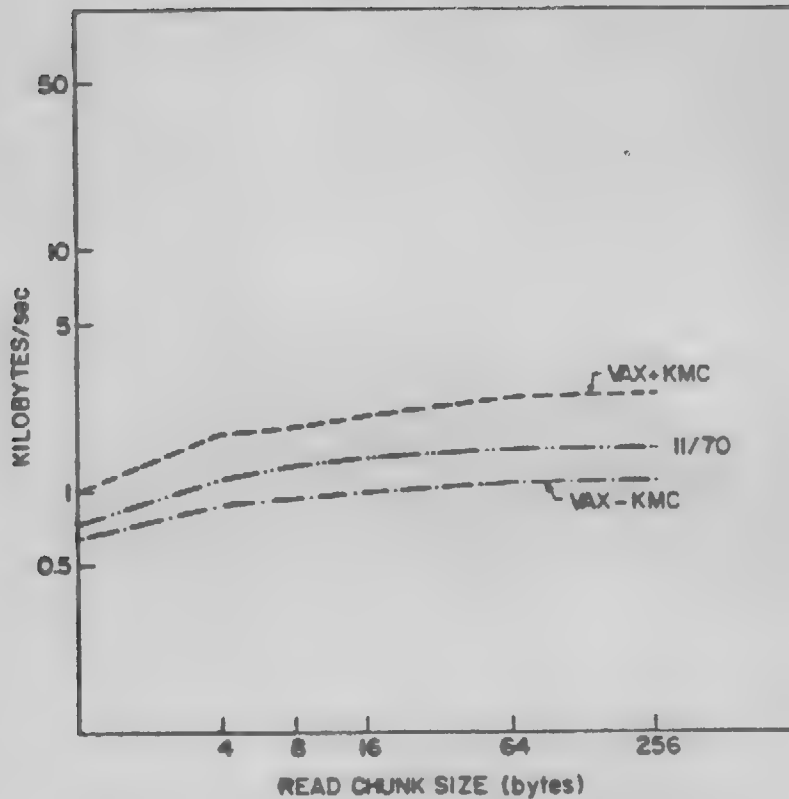


Figure 4. Terminal Data Loop-Around - Cooked Mode

The VAX-11/780 yielded a gain of 26%, while the PDP-11/70 showed a slight decrease of 3%. Table 7 throughput gains translate into corresponding increases in the number of users that can be supported in a program development environment.

### 3.5 Administrative Guidelines

The number of timesharing users that can be supported depends on a number of factors including the type of work, fraction of users active as opposed to simply logged-on, quality of service desired and willingness to limit heavy resource consuming activities. The following guidelines for typical program development environments are based on interviews with computer center customers with expertise in machine sizing; they are, necessarily, rough estimates.

Experience indicates that approximately thirty-five users logged-in and active gives a reasonable response time compromise for VAX-11/780 System V systems with 3 disk drives. VAX-11/750 and PDP-11/70 machines will support about two-thirds the load of the VAX-11/780, assuming two drives for the VAX-11/750 and three for the PDP-11/70. Use of higher loading is feasible with activity restrictions and willingness to tolerate poor response time. On the other hand, lighter loadings will yield improved response. The ordinate in Figure 5 is a measure of system capacity; experience shows that 10,000 jobs/hour in this figure corresponds to roughly thirty-five (reasonably) satisfied users.

Throughput in Table 7 for System V on the VAX-11/780 is for a system configured with two hundred system buffers. Installations with two or more megabytes of memory may get a performance increase of a few percent relative to Table 7 by configuring with a larger number of buffers.

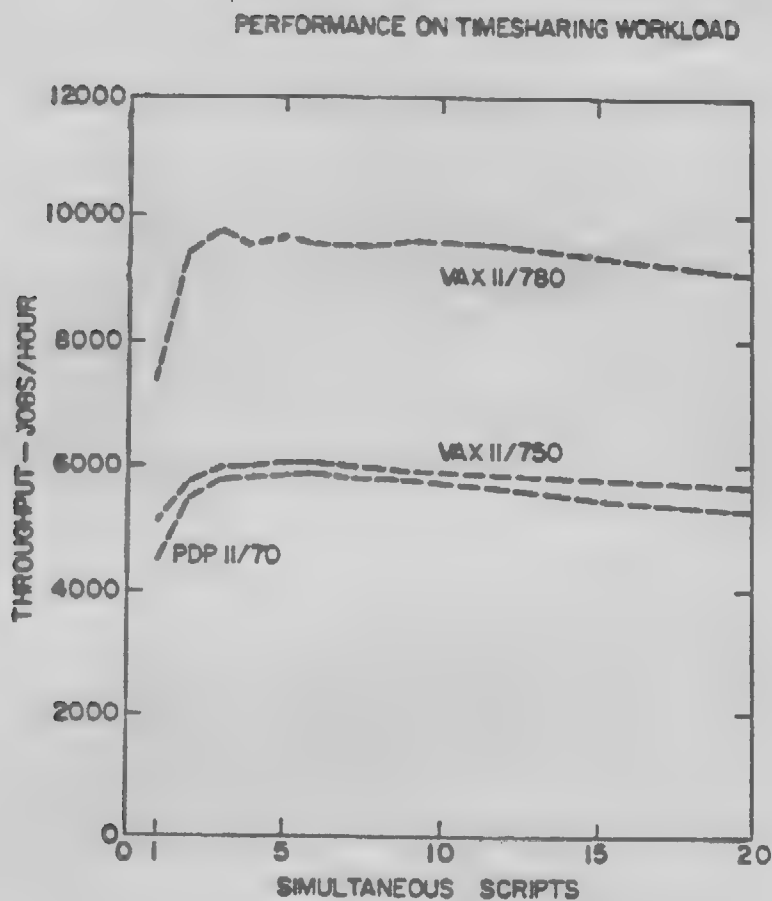


Figure 5. UNIX System V Throughput vs. Applied Load

System buffers and user programs both contend for main memory, so that the optimum number of buffers is a compromise. The number of buffers may be increased until swapping overhead becomes significant. The administrator can obtain the optimal balance for a particular system by monitoring system activity using *sar(1)* and reducing the number of configured buffers when a significant increase in swapping is observed.

The results presented reflect the improved performance obtained by setting the "sticky" bits on the benchmark commands. System administrators should set the "sticky" bits on the commands that are heavily used at their particular installation. Results with "sticky" bits off were five to ten percent poorer.

#### 4. DOCUMENTATION

##### 4.1 Published Documents

There are two major parts to the standard documentation package: the guides, and the manuals. The following is a list of the guides and manuals applicable to *System V*:



*System V Documentation*

- UNIX System Release Description—System V
- UNIX System Transition Aid—System V
- UNIX System User's Manual—System V
- UNIX System Administrator's Manual—System V
- UNIX Operating System Error Message Manual—System V
- UNIX System Administrator's Guide
- UNIX System Operator's Guide
- UNIX System User's Guide
- UNIX System Document Processing Guide
- UNIX System Graphics Guide
- UNIX System Programming Guide
- UNIX System Support Tools Guide

One copy of these documents will be provided to the purchaser of a UNIX System license. Additional copies may be ordered through the sales contact at AT&T Technology and Licensing.

**4.2 Interim Documents**

Additional documentation not yet incorporated in the other *System V* manuals or guides is being distributed in the appendices of this *SRD*.

**4.2.1 New Manual Pages**

*Appendix A* provides new manual pages for the *UNIX System User's Manual* and the *UNIX System Administrator's Manual*. These new manual pages are not found in the published UNIX System manuals.

The manual pages of special interest are highlighted here. See the manual pages *ex(1)*, *vi(1)* and *termcap(5)* for the new text editor *vi*. *Fsba(1M)* is a new manual page for a tool used in converting to a 1K block file system.

**4.2.2 Setting Up the UNIX System**

The *UNIX System Administrator's Guide* contains a section for loading the system tapes and setting up a new UNIX system. The new *init* and *getty* programs have invalidated the installation procedures in this section. **A revised section which must be used for setting up the DEC systems during the installation of System V is provided in *Appendix B*.**

**4.2.3 UNIX System V Init and Getty**

The memorandum *UNIX System V Init and Getty* is provided in *Appendix C*. This memorandum describes the interface and features of the new *init* and *getty* programs. It provides useful transitional information along with the corresponding manual pages in the *UNIX System Administrator's Manual*.

**4.2.4 Screen Editors**

Two documents have been included for the new *vi(1)* and *ex(1)* commands. *An Introduction to Display Editing with Vi (Appendix D)* is a tutorial introduction to the screen editor. It also contains a summary table of all the commands for reference. The *Ex Reference Manual (Appendix E)* describes the line oriented text editor.

**4.2.5 Uucp Documentation**

*Appendix F* contains the *Uucp Tutorial* and *Appendix G* contains the *Uucp Administrator's Manual*. These two documents describe how to set up and use the *uucp(1C)* system.

**4.2.6 ACU Installation**

*Appendix H* contains the memorandum *Installing an Auto Call Facility on a UNIX System*. This information should be especially useful for setting up Auto Call Units (ACU) to be used with *uucp(1C)* and *cu(1C)*.

#### 4.2.7 Manual Pages

The manual pages for the published version of the *UNIX System User's Manual* and the *UNIX System Administrator's Manual* are contained on the system in the directories `/usr/man/u_man` and `/usr/man/a_man`. The manual pages provided in *Appendix A* are to be found in the directory `/usr/man/local`. This arrangement allows the `man(1)` command to access the latest manual page while keeping the published manual pages separated from the updated pages. The system administrator may wish to move these updated manual pages from `/usr/man/local` to their usual home in either `/usr/man/u_man` or `/usr/man/a_man` at installation time so that the site local manual pages can be placed in `/usr/man/local`.

The pages for the new *UNIX Operating System Error Message Manual* are contained in the directory `/usr/docs/errmanual`. This manual is not obtainable by use of `man(1)`. However, `troff(1)`, `nroff(1)`, and `man(5)` can be used to obtain the formatted manual pages.

### 5. ITEMS NO LONGER SUPPORTED

#### 5.1 DQS and Fget

The list of items no longer supported includes:

- DQS (on the PDP-11/70)
- `fget.demon` and `fget.odemon`

#### 5.2 Floating Point Simulation

Floating point simulation is no longer supported for any processor. This means *System V* will only run on PDP processors with floating point hardware.

#### 5.3 DEC Processors

DEC processors which are no longer supported include:

- PDP-11/23
- PDP-11/34
- PDP-11/45

### 6. TRANSITIONAL INFORMATION

The information provided here is intended to ease the transition from *System III* to *System V*. Additional information is provided in the document *UNIX System Transition Aid - System V*, which should also be consulted.

The transitional information included here is applicable to all supported processors unless otherwise noted.

#### 6.1 Release Tapes

This release is being provided as a set of initial load tapes for each of the supported VAX and PDP processor lines. The following tapes constitute the initial load tapes provided for *System V*:

Tapes for the VAX-11/780, VAX-11/750, and PDP-11/70:

- Boot Loader + *root* File System + Selectables (Tape 1)
- User File System (Tape 2)

The detailed format and contents of these release tapes can be found in the document *Setting up the UNIX System (DEC)*. See *Appendix B*.

#### 6.2 Installation Documentation

Installation procedures that have been revised since their last formal publication are included in *Appendix B*. This revised document must be used to install this release. **Do not use the "setting up" procedures in the published *UNIX System Administrator's Guide* to install the system.**

### 6.3 File System Conversion (VAX only)

The procedures for converting to a 1K block file system are provided in the document *UNIX System Transition Aid - System V*. A tool to aid in the conversion, the file system block analyzer *fsba(1M)* is provided in *System V*.

### 6.4 Init & Getty Interface

The new *init* and *getty* are described in *Appendix C*. This document and the associated manual pages (*init(1M)*, *getty(1M)*, *gettydefs(4)*, *inittab(4)*, *brc(1M)*, and *shutdown(1M)*) should be read before attempting to install *System V*.

When making changes to the *shutdown* procedure (e.g., while "setting up" the system), make sure the *telinit* command in the *shutdown* procedure is the last command in that procedure; i.e., no other commands can follow *telinit* for this procedure to work correctly.

### 6.5 C Programs

The handling of enumeration data types has been corrected. Previously an *enum* variable would be stored in the smallest data type that would hold it; now it is always stored in an integer. This makes newly compiled object files incompatible with those compiled with a UNIX System before *System V*. Thus, all local library functions (and other object modules) employing enumeration data types must be recompiled to prevent variable size incompatibilities when they are link-edited (*ld(1)*).

The following change has been made to the *scanf(3S)* functions. If a "%" specification fails to match an input character, the function returns with a mismatch. In the previous version, this was treated as a match and a string consisting only of the *NUL* character was returned. The new behavior conforms to that of other UNIX System implementations.

Programs that include *utmp.h* must be recompiled. Some structure definitions in *utmp.h* have been changed. Some code changes may also be required.

On the VAX, if one of the source files for a program is changed, a re-compilation of all source files will be required before attempting to link edit, *ld(1)*, the object files. This is necessary because of the implementation of the new *Common Object File Format*.

### 6.6 Vi

If *make(1)* is used to rebuild the *vi(1)* command and it terminates prematurely for any reason, the following precaution must be taken. The "make" of *vi(1)* must be restarted only after running the command

make clean

This is because all error messages and strings are extracted from the source code and placed in a common library of messages and strings. This library resides in */usr/lib/ex3.7strings*, once the *make* has completed. There is a good chance this library will be corrupted if the *make* terminates prematurely.

### 6.7 Uucp

A new feature of the *uucp(1C)* command is the ability to specify a routing sequence within a source or destination file argument. This forwarding feature allows files to be passed through several machines before arriving at the destination machine. Users must be aware that this feature will be blocked by any machine in the routing list that is not set up to allow forwarding and that is not running *System V*.

When setting up *uucp(1C)* while installing *System V*, the files */usr/lib/uucp/L\_stat* and */usr/lib/uucp/R\_stat* must be removed. These files have changed from ASCII to binary. If they are not removed, the *uustat(1C)* command will fail.

The names of the *uucp(1C)* work files were increased by one character to provide job tracking via *uustat(1)*. *Uucp(1C)* will not recognize old work files. Therefore, these files (C.\*) must be removed from the spool directory */usr/spool/uucp*.

The `-m` option will not do anything when `uucp(1C)` is executed via the `-e` option (remote execution option). `Uulog(1C)` no longer produces any output unless an option is specified on the command line. `Uuclean(1M)` should be used on valid `uucp(1C)` spool directories only.

A tutorial on `uucp(1C)` and a manual on the usage and administration of a `uucp` network are included as appendices to this document, and should be consulted for other details. See *Appendix F* and *Appendix G*.

### 6.8 PDP Size Limitations

The address space in the operating system on the PDP processors is limited. Loading drivers, such as VPM, may require changes to the operating system makefiles. See the *Setting Up the UNIX System (DEC)* document in *Appendix B* for details.

### 6.9 Modification Requests

A summary of UNIX System modification requests (MRs) resolved in *System V* is provided in *Appendix I*.

## 7. KNOWN DEFICIENCIES

The following deficiencies are known to exist in *System V* and are itemized here to allow the user to avoid them until they are fixed in a future UNIX System release.

### 7.1 Compilers and Language Tools

#### 7.1.1 C Compilers

The compiler currently does not detect a shortage of temporary file space in the `/tmp` or `/usr/tmp` directory and will report compilation errors when this condition occurs. This situation will be diagnosed properly in the next release. The user can specify a directory for temporary file space by setting and exporting the environment variable `TMPDIR`.

On the VAX systems, if an unsigned integer or unsigned long integer with the high order bit set is converted to **double** or **float**, a negative value is produced instead of a large positive value. This will be fixed in the next release.

On the VAX systems, casting a *float* or *double* to *long*, *int*, *short*, etc., followed by a mask, will not be optimized correctly if the mask consists of contiguous low order bits. For example:

```
double d = 1.7;
long l = ((long)d) & 0x7FFFFFFL;
```

On the VAX systems, when using bit fields, if one of the fields happens to be the high-order 16 bits of a word, accesses of the field will be optimized incorrectly. For example:

```
struct
{
    unsigned low:16;
    unsigned high:16; /* field will be accessed wrong */
} x;
```

For compilers or cross-compilers running on the PDP-11/70 processor, structures and **automatic** arrays are limited in size to fewer than 4096 bytes.

#### 7.1.2 F77 Compiler

On the VAX systems, a relational expression with the integer constant "0" on the left hand side (e.g., "if (0 .gt. i) ...") is optimized incorrectly. To eliminate the problem, reverse the expression so the "0" is on the right-hand side. This will be fixed in the next release.

On the VAX systems, the `-o` option doesn't work as documented. It must appear after the last filename on the command line, not before.

### 7.1.3 Machine Language Processors

On the VAX systems, the `-e` option for the link editor `ld(1)` doesn't work. These problems will be dealt with in the next release.

On the VAX systems, if `strip(1)` is to be run against an archive that has been built from executable files (as opposed to unlinked ".o" files), the `-r` option must be used. Failure to do so will damage the archive without a warning. Also, the little used options `-l`, `-s` and `-x` do not work properly and should not be used. These problems will be corrected in the next release.

### 7.1.4 Libraries

The `abort(3C)` function doesn't close (flush) files, so some output may be left in internal buffers. This will be changed in the next release.

The values returned by `ftell(3S)` on a file opened for appending (i.e., with type `a` or `a+`) may be confusing. For further information, see the manual page for `fopen(3S)`.

The value returned by `log(3M)` and `log10(3M)` on non-positive input is `-HUGE`, not 0 as documented.

If a program opens more than one stream simultaneously with `popen(3S)`, it must close them in the reverse order (most recently opened first).

The commands "`cc -p ...`" on the PDP and "`f77 -p ...`" on all supported processors do not automatically load the profiled versions of the libraries. They must be named explicitly on the command line: `/lib/libp/libm.a` (Math Library) and `/lib/libp/libc.a` (C Library).

### 7.1.5 Debugging Tools

The `lint(1)` analyzer has trouble with `enums` as function arguments.

The profiler `prof(1)` fails if the program text is larger than 64K bytes. This limitation will be removed in the next release.

The cross-referencer `cxref(1)` considers each appearance of a formal parameter in a macro definition as a definition of that parameter which may therefore appear many times. Also, the temporary files created may exceed the system limit, leading to incomplete cross-reference listings. The program will diagnose this failure clearly in the next version.

## 7.2 Communications

### 7.2.1 Uucp

The `-r` option is missing from the `uucp(1C)` manual page. This option delays the execution of the `uucico` daemon process, and is useful in `shell` scripts that invoke many `uucp(1C)` commands. The option should be specified on all but the last `uucp(1C)` command so that only one invocation of `uucico` occurs.

### 7.2.2 Uuname

If the entry for a system in `/usr/lib/uucp/L.sys` exceeds 128 characters, the output of `uuname(1)` will contain some extraneous information which can be used to compromise system security. A solution to the problem is to re-make `/bin/uuname`, after editing `/usr/lib/cmds/uucp/uuname.c` to increase the size of the array `s` to 300 (from 128), and increase read byte limit of the `fgets(3)` subroutine calls to 300 (from 128). The `make` file for `uuname(1)` is `/usr/lib/cmds/uucp/uucp.mk`.

### 7.2.3 Cu

The `cu(1C)` manual page will be altered in the next release to clarify the use of the `-s` and `-l` options. When using a dial line, the speeds available using a 212A data-set are 300 and 1200 baud; the default speed is 300. If a speed is specified using the `-s` option and if it is less than or equal to 300, 300 will be used; if the speed is greater than 300, then 1200 will be used. When a direct line is used, the speed specified by the `-s` option is used. If the `-l` option is used, the `-s` option is ignored and the speed is found in the `L-devices` file for the corresponding line.

The manual page will be changed in the next release to restate the meaning of the input sequence `~%nostop`. Currently the manual page states that `~%nostop` command turns off the input control for the remainder of the terminal session. This is incorrect. The `~%nostop` command toggles between DC1/DC3 input control and no input control.

#### 7.2.4 VPM

If the `vpmstart(1M)` command is included in `/etc/brc`, then the error message "getopt: not found" will be printed on the system console when the system is booted. This is the result when `vpmstart` attempts to invoke `getopt(1)` before the `/usr` file system is mounted (`getopt(1)` is normally located in `/usr/bin`). To avoid this problem, move (`mv(1)`) `getopt` from `/usr/bin` to `/bin`.

#### 7.3 Lp

If the user attempts to print a non-ASCII file using `lp(1)`, the result will be an error message of the form "xxxx Bus Error - core dumped." In the next release, the `lp(1)` program will be modified to filter (remove) non-ASCII characters.

If an error message about the file `outputq` results when attempting to use `lp(1)`, perform the following file creation command:

```
cp /dev/null /usr/spool/lp/outputq
```

A problem exists that may occasionally result in the wrong file being printed. If a user tries to print a file that happens to have the same name as a file in the spool directory (`/usr/spool/lp`), then `lp(1)` will print the file from its own directory instead of the users file. Temporarily rename your file to print it.

The `hp2631a` filter distributed with `lp(1)` randomly drops characters from the printout. This can be cured by changing `termio(4)` options in the filter program by adding the following line:

```
tbuf.c_cc[VMIN]=1;
```

If the error message "Model non-existent" occurs when a user tries to copy a model interface program using the `lpadmin(1M)` command, the files in the `/usr/spool/lp/model` directory should have the following permissions, owner, and group:

```
Permissions: -rw-r--r--
Owner: lp
Group: bin
```

The `accept(1M)` and `reject(1M)` commands always return a successful indication even when they fail.

#### 7.4 Init and Getty

`Init(1M)` uses the pathname `/dev/syscon` (which is normally linked to `/dev/console`) for output to the console. If "init s" is executed from a terminal (e.g., in `shutdown(1M)`) other than `/dev/console`, `/dev/syscon` will be linked to *that* terminal. Typing the `delete` key from `/dev/console` returns control to `/dev/console` only if the *remote* console has hung up.

#### 7.5 File

While using `file(1)` to determine the type of an object file on the PDP-11/70, it will be inaccurately reported that the file is "4.0 pure executable". It should be reported that the file is "5.0 pure executable". The problem is associated with the `file(1)` command, not the object file.

#### 7.6 Du and Ls

The size, in blocks, of a file or file system can be incorrectly reported by `du(1)` and `ls(1)` (using the *-s option*). Sizes returned by these commands are multiples of 512-byte blocks. Therefore, for 1024-byte block files or file systems, sizes should always be even numbers. However, a one block error in the size can occur because of a flaw in the algorithm used to determine the size.

7.7 /etc/checkall

The prototype /etc/checkall procedure included on the System V installation tape is incorrect. The -T option to be passed to *dfsk*(1M) should be -t (lower case).

7.8 Mm Macros

The System V *mm*(1) macros have Bell Laboratories specific features which should be removed. The *non-btl*(1) command is provided by System V to re-install the *mm*(1) macros without these specific features. A manual page for *non-btl*(1) is provided in Appendix A.

7.9 Games

The set of games provided in the /usr/games directory is not consistent with those described in the UNIX System User's Manual, nor is the available set of games identical across processor lines. Table 8 attempts to clarify this confusion. It identifies the games available on each processor line, the existence of a source file, and the availability of a manual page. For each game whose source exists, the executable game was built from this source.

The /usr/games directory also contains a subdirectory called *lib* and two commands (*turnon* and *turnoff*) to enable and disable the use of the games. Note there is no longer a *login* entry for games in the distributed /etc/passwd file.

TABLE 8. State of /usr/games

Game	Source?	Manpage?	PDP?	VAX?
adventure	no	no	yes	yes
arithmetic	yes	yes	yes	yes
back	yes	yes	yes	yes
biorhythm	no	no	yes	no
bj	yes	yes	yes	yes
checkers	no	no	yes	no
chess*	no	yes	yes	no
craps	yes	yes	yes	yes
cubic	no	yes†	yes	no
fish	yes	no	yes	yes
fortune	yes	no	yes	yes
hangman	yes	yes	yes	yes
jotto	no	yes	yes	yes
mastermind	yes	no	yes	yes
maze	yes	yes	yes	yes
monopoly	no	no	yes	no
moo	yes	yes	yes	yes
number*	yes	no	yes	yes
psych*	yes	no	yes	yes
quiz	yes	yes	yes	yes
random*	yes	no	yes	yes
reversi	no	yes	yes	no
sky	no	yes	no	no
snake*	no	no	yes	yes
trk*	yes	no	yes	yes
ttl	yes	yes	yes	yes
wump	yes	yes	yes	yes

\* Needs instructions.

† Not in System V manual; on-line page only.

## 8. DISCLAIMERS

This document includes specific reference to the use of the UNIX System on a particular processor, the Western Electric Company 3B20S, which is not presently available except for internal use within the Bell System. Some commands, system calls, subroutines, file formats, and special files described in the *System V* documentation and associated with the 3B20S, are not included with the release of *System V*. Manual pages associated with them are marked with *(3B20S only)* at the top of the page.

Additional commands, subroutines, and special files described in the *System V* documentation are not included with the release of *System V*. Manual pages associated with them are included in the list:

<i>nscloop</i> (1M)	<i>nscmon</i> (1M)	<i>nscstat</i> (1C)	<i>nsctorje</i> (1C)
<i>nusend</i> (1C)	<i>scat</i> (1)	<i>se</i> (1)	<i>stgetty</i> (1M)
<i>st</i> (1M)	<i>sstat</i> (1)	<i>stlogin</i> (1)	<i>trouble</i> (1)
<i>x25lnk</i> (1C)	<i>x25pvc</i> (1C)		
<i>x25alnk</i> (3C)	<i>dial</i> (3C)	<i>x25clnk</i> (3C)	<i>x25dlnk</i> (3C)
<i>x25hlnk</i> (3C)	<i>x25ilnk</i> (3C)	<i>x25ipvc</i> (3C)	<i>x25rpvc</i> (3C)
<i>nsc</i> (7)	<i>st</i> (7)	<i>stermio</i> (7)	<i>x25</i> (7)

Some system maintenance procedures included in the *System V* documentation are not intended for use with *System V*. Manual pages associated with them include *trouble*(8), and other manual pages, marked *(3B20S only)*, in *Section 8* of the *UNIX System Administrator's Manual*.